**ĐẠI HỌC ĐÀ NẴNG**

**ĐẠI HỌC BÁCH KHOA**

**KHOA CÔNG NGHỆ THÔNG TIN**



**PBL2: DỰ ÁN CƠ SỞ LẬP TRÌNH**

**Tên đề tài :** *Quản lý nhân viên*

SINH VIÊN THỰC HIỆN:

Tên sinh viên 1: **HOÀNG NGUYỄN NHẬT MINH**

Lớp: **21TCLC\_NHAT2** Nhóm: **21NH99**

Tên sinh viên 2: **NGUYỄN VĂN DŨNG**

Lớp: **21TCLC\_NHAT2** Nhóm: **21NH99**

GIẢNG VIÊN HƯỚNG DẪN: **ThS. ĐỖ THỊ TUYẾT HOA**

**Đà Nẵng 12-2022**

LỜI MỞ ĐẦU

Việc áp dụng các phần mềm tin học vào các lĩnh vực giúp nâng cao tính hiệu quả và chính xác của công việc, ngoài ra còn tiết kiệm thời gian và giảm bớt mệt nhọc cho con người khiến hệ thống công việc hoạt động nhịp nhàng hơn. Các công ty luôn luôn phát triển, các hồ sơ tuyển vào sẽ nhiều lên vì vậy đòi hỏi phải cần rất nhiều những kho chứa hồ sơ để lưu trữ hồ sơ của nhân viên khi vào công ty. Để giảm đi những bất cập trong công tác quản lý nhân sự và lương trong công ty, giải pháp hiệu quả nhất hiện nay là ứng dụng công nghệ thông tin, tự động hoá trong điều hành và quản lý nhân sự để nâng cao chất lượng phục vụ, đổi mới phương thức quản lý, tăng năng xuất hiệu quả. Nhận thấy được nhu cầu ấy, chúng em đã chọn đề tài cho “Dự án Cơ sở lập trình” là “Quản Lý Nhân Sự”. Cùng với tốc độ phát triển mạnh mẽ của Công nghệ thông tin các lập trình viên đã phát minh ra nhiều phần mềm hữu ích nhằm phục vụ cho công việc của con người và phần mềm quản lý nhân sự cũng là một trong những vấn đề con người quan tâm nhiều nhất, nó giúp cho công tác nghiệp vụ của các công ty giảm thiểu tối đa những vất vả trong công việc giúp cho việc lưu trữ hồ sơ dễ dàng hơn, giảm thiểu diện tích các kho và thậm chí là không cần. Đề tài “Quản Lý Nhân Sự” là một đề tài không phải là mới, hiện tại trên thị trường cũng có rất nhiều các chương trình quản lý nhân sự có chức năng tương tự và chuyên nghiệp, nhưng vì mong muốn có thể tìm hiểu một phần kiến thức nhỏ nhoi trong một sa mạc công nghệ thông tin nên chúng em đã đi vào tìm hiểu về cơ cấu tổ chức và phát triển công ty để viết ra một chương trình quản lý nhân sự. Qua đó chúng em cũng sẽ hiểu rõ hơn về cách xây dựng, vận hành một chương trình quản lý. Chương trình của chúng em được lập trình bằng ngôn ngữ lập trình C++. Chương trình này có nhiều chức năng như quản lý hồ sơ, lương, chế độ của nhân viên giúp giảm tải bớt việc tìm kiếm mất thời gian mà độ hiệu quả và tính chính xác vẫn ở mức cao.

MỤC LỤC

[LỜI MỞ ĐẦU 2](#_Toc120657418)

[MỤC LỤC 3](#_Toc120657419)

[**DANH MỤC HÌNH VẼ** 5](#_Toc120657420)

[1. GIỚI THIỆU ĐỀ TÀI 6](#_Toc120657421)

[1.1. Giới thiệu vấn đề 6](#_Toc120657422)

[1.2. Mục tiêu đề tài. 6](#_Toc120657423)

[1.3. Ý nghĩa khoa học và thực tiễn của đề tài. 6](#_Toc120657424)

[2. PHÂN TÍCH CHỨC NĂNG HỆ THỐNG 7](#_Toc120657425)

[2.1. Chức năng “Hệ Thống” 7](#_Toc120657426)

[2.2. Chức năng “Quản lý tiền lương” 8](#_Toc120657427)

[3. THIẾT KẾ CẤU TRÚC DỮ LIỆU 8](#_Toc120657428)

[3.1. Phát biểu bài toán 8](#_Toc120657429)

[3.2. Phân tích và ứng dụng cấu trúc dữ liệu trong hệ thống 10](#_Toc120657430)

[3.2.1. Danh sách liên kết đơn ( Linked List ) 10](#_Toc120657431)

[3.2.2. Bảng băm ( HashTable ) 16](#_Toc120657432)

[4. PHÂN TÍCH HƯỚNG ĐỐI TƯỢNG VÀ TRIỂN KHAI HỆ THỐNG 20](#_Toc120657433)

[4.1. Cấu trúc hệ thống hướng đối tượng 20](#_Toc120657434)

[4.1.1. Quản lý nhân viên ( class ListEmployee ) 21](#_Toc120657435)

[4.1.2. Quản lý chức vụ ( class ListPosition ) 25](#_Toc120657436)

[4.1.3. Quản lý phòng ban ( class ListMD ) 28](#_Toc120657437)

[4.2. Kết quả 30](#_Toc120657438)

[4.2.1. Giao diện chính của chương trình 30](#_Toc120657439)

[4.2.2. Kết quả thực thi của chương trình 30](#_Toc120657440)

[4.2.3. Nhận xét 37](#_Toc120657441)

[5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 37](#_Toc120657442)

[5.1. Kết luận 37](#_Toc120657443)

[5.1.1. Tóm tắt những kết quả đạt được. 37](#_Toc120657444)

[5.1.2. Những mặt còn hạn chế. 37](#_Toc120657445)

[5.2. Hướng phát triển 38](#_Toc120657446)

[TÀI LIỆU THAM KHẢO 39](#_Toc120657447)

**DANH MỤC HÌNH VẼ**

[*Hình 3.1.1: Minh họa về file lưu trữ thông tin nhân viên* 9](#_Toc122116497)

[*Hình 3.1.2: Minh họa về file lưu trữ thông tin theo chức vụ* 10](#_Toc122116498)

[*Hình 3.1.3: Minh họa về file lưu trữ thông tin theo phòng ban.* 10](#_Toc122116499)

[*Hình 3.1.4: Minh họa về file lưu trữ thông tin làm việc của nhân viên* 11](#_Toc122116500)

[*Hình 3.2.1: Hình minh họa danh sách liên kết đơn* 12](#_Toc122116501)

[*Hình 3.2.2: Hình class LinkedList được áp dụng trong chương trình.* 12](#_Toc122116502)

[*Hình 3.2.3: Mô hình bảng băm (HashTable)* 17](#_Toc122116503)

[*Hình 3.2.4: Class HashTable được áp dụng trong bài toán* 18](#_Toc122116504)

[*Hình 4.1.1: Mô hình class được áp dụng trong chương trình* 22](#_Toc122116505)

[*Hình 4.1.2: Mô hình class Person được áp dụng trong chương trình.* 23](#_Toc122116506)

[*Hình 4.1.3: Mô hình class Employee được áp dụng trong chương trình* 24](#_Toc122116507)

[*Hình 4.1.4: Mô hình class ListEmployee được áp dụng trong chương trình* 25](#_Toc122116508)

[*Hình 4.1.5: Mô hình class Position được áp dụng trong chương trình* 27](#_Toc122116509)

[*Hình 4.1.6: Mô hình class ListPosition được áp dụng trong chương trình* 28](#_Toc122116510)

[*Hình 4.1.7: Mô tả class MD được áp dụng trong chương trình* 29](#_Toc122116511)

[*Hình 4.1.8: Mô tả class ListMD được áp dụng trong chương trình* 30](#_Toc122116512)

[*Hình 4.2.1: Giao diện chương trình khởi đầu* 31](#_Toc122116513)

[*Hình 4.2.2: Giao diện đăng nhập của chương trình* 31](#_Toc122116514)

[*Hình 4.2.3: Giao diện của sổ lựa chọn các chức năng của chương trình* 31](#_Toc122116515)

[*Hình 4.2.4: Giao diện chức năng in danh sách* 32](#_Toc122116516)

[*Hình 4.2.5: In danh sách nhân viên có trong nhà hàng* 32](#_Toc122116517)

[*Hình 4.2.6: Giao diện chức năng sắp xếp* 32](#_Toc122116518)

[*Hình 4.2.7: Sắp xếp nhân viên theo thứ tự tăng dần của mã số nhân viên* 33](#_Toc122116519)

[*Hình 4.2.8: Giao diện chức năng xóa* 33](#_Toc122116520)

[*Hình 4.2.9: Xóa một phòng ban ra khỏi danh sách* 34](#_Toc122116521)

[*Hình 4.2.10: Giao diện chức năng thêm trong danh sách* 34](#_Toc122116522)

[*Hình 4.2.11: Thêm một nhân viên mới vào trong danh sách* 35](#_Toc122116523)

[*Hình 4.2.12: Giao diện chỉnh sửa thông tin trong danh sách* 35](#_Toc122116524)

[*Hình 4.2.13: Chỉnh sửa thông tin của một nhân viên trong danh sách* 36](#_Toc122116525)

[*Hình 4.2.14: Giao diện chức năng tìm kiếm trong chương trình* 36](#_Toc122116526)

[*Hình 4.2.15: Tìm kiếm nhân viên mã phòng ban.* 37](#_Toc122116527)

[*Hình 4.2.16: Giao diện thông kê nhân viên* 37](#_Toc122116528)

[*Hình 4.2.17: Chức năng thống kê được áp dụng trong chương trình* 38](#_Toc122116529)

# GIỚI THIỆU ĐỀ TÀI

## Giới thiệu vấn đề

Tại các công ty, doanh nghiệp, quản lý nhân sự là một trong những vấn đề cần thiết và cấp bách. Một công ty muốn có được thành tích tốt trong kinh doanh và đứng vững trên nền kinh tế hiện nay thì cần có một số phần mềm riêng cho công ty của mình.

Vì vậy việc áp dụng Công nghệ thông tin để quản lý công ty là việc rất cần thiết, giúp cho người quản lý dễ dàng quản lý thông tin và tra cứu thông tin một cách nhanh chóng và chính xác. Tiết kiệm được nhiều thời gian, công sức cho tất cả mọi người.

Quản lý nhân sự trong tổ chức luôn là điều rất quan trọng thậm chí là điều then chốt trong xã hội công nghệ 4.0 hiện nay. Bởi vậy hoạt động quản lý nhân sự càng trở nên cần thiết hơn bao giờ hết. Quản lí trên giấy tờ sẽ tốn thời gian, công sức, dễ mất giấy tờ đã lưu lại, khó thống kê báo cáo. Vì vậy cần phải có phần mềm để quản lí nhân sự để giúp người dùng dễ quản lí, nhanh chóng, chính xác và tiết kiệm nhiều thời gian…

Từ những ý trên, với mong muốn áp dụng Công nghệ thông tin vào quản lý, vì vậy nên chúng em đã xây dựng chương trình “Quản Lý Nhân Sự”, với các chức năng lưu trữ và xử lý thông tin chính xác và nhanh chóng.

## Mục tiêu đề tài.

* Nghiên cứu, xây dựng, vận hành một chương trình quản lý hoàn chỉnh

với nhiều chức năng tối ưu để có thể áp dụng vào thực tế sử dụng.

* Áp dụng lập trình hướng đối tượng OOP, sử dụng ngôn ngữ lập trình

C++ cùng các cấu trúc dữ liệu vào việc xây dựng một chương trình quản lý.

## Ý nghĩa khoa học và thực tiễn của đề tài.

* Ý nghĩa khoa học:

Kết quả nghiên cứu của đề tài góp phần nâng cao nhận thức, khả năng thu thập thông tin, phân tích yêu cầu của người dùng để xây dựng được chương trình quản lý. Bước đầu tiếp cận ngôn ngữ lập trình để xây dựng chương trình.

* Ý nghĩa thực tiễn:

Việc chọn đề tài “Xây dựng ứng dụng quản lý nhân sự” là một công việc có tính khả thi cao và khả năng ứng dụng thực tế hiệu quả. Đề tài nếu ứng dụng thực tế có hiệu quả sẽ góp phần cải thiện hệ thống “Quản lý nhân sự”.

# PHÂN TÍCH CHỨC NĂNG HỆ THỐNG

## Chức năng “Hệ Thống”

Ở chức năng này người trực tiếp quản lý sẽ thực hiện được những yêu cầu sau:

* + - In danh sách thông tin nhân viên bao gồm:
      * Các thông tin cơ bản như: Mã số nhân viên, Họ tên, Ngày sinh, CCCD, Giới tính, Số điện thoại, Địa chỉ.
      * Danh sách các chức vụ bồm gồm các thông tin như: Mã chức vụ, Tên chức vụ, Phụ cấp, Mức lương quy định của từng chức vụ.
      * Danh sách phòng ban bao gồm: Mã phòng ban, Tên phòng ban.
      * Danh sách vị trí của từng nhân viên bao gồm: Mã nhân viên, Tên nhân viên, Chức vụ hiện tại của từng nhân viên, Phòng ban hiện tại của từng nhân viên.
    - Sắp xếp nhân viên theo lựa chọn:
      * Sắp xếp nhân viên theo mã số nhân viên tăng dần hoặc giảm dần và in ra danh sách được sắp xếp.
      * Sắp xếp nhân viên theo số tiền lương nhận được trong tháng tăng dần hoặc giảm dần và in ra danh sách được sắp xếp.
    - Xóa nhân viên nghỉ làm hoặc xóa chức vụ hoặc phòng ban nào đó trong trường hợp bị giải thể.
    - Thêm nhân viên, chức vụ hoặc phòng ban mới vào danh sách.
    - Cập nhật tất cả các thông tin của nhân viên bao gồm:
      * Các thông tin cơ bản như: Mã số nhân viên, Họ tên nhân viên, Ngày sinh, CCCD, Giới tính, Số điện thoại, Địa chỉ.
      * Các thông tin về chức vụ như: Mã chức vụ, Tên chức vụ, Phụ cấp và mức lương quy định của từng chức vụ.
      * Các thông tin về phòng ban như: Mã phòng ban, Tên phòng ban.
    - Tra cứu thông tin nhân viên bao gồm:
      * Tra cứu các thông tin cơ bản của từng nhân viên theo mã nhân viên.
      * Tra cứu nhân viên theo từng chức vụ.
      * Tra cứu nhân viên theo từng phòng ban.
      * Tra cứ chức vụ, phòng ban và chức vụ nào thuộc phòng ban nào.
    - Thông kê nhân viên theo các thông tin như:
      * Thông kê số lượng nhân viên thuộc từng chức vụ trong từng phòng ban.
      * Thông kê có bao nhiêu nhân viên nam, nữ làm việc thuộc từng chức vụ, trong từng phòng ban.
      * Thông kê tổng số thu nhập của nhân viên theo từng phòng ban.

## Chức năng “Quản lý tiền lương”

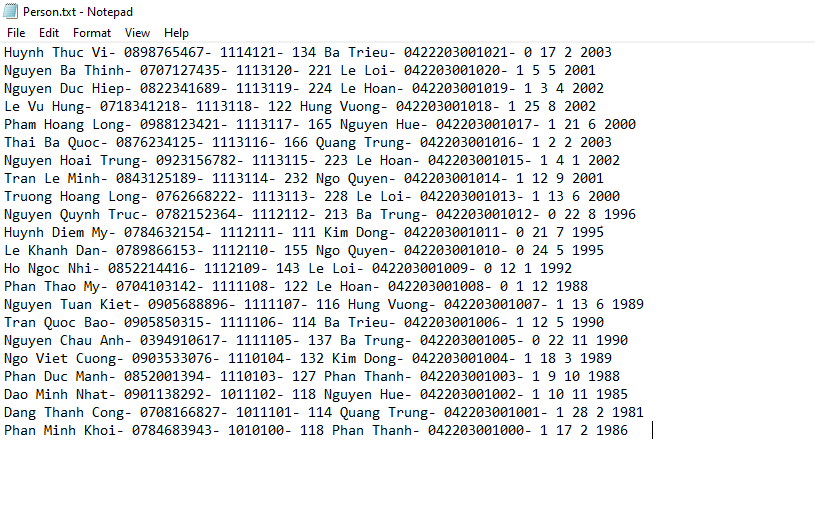
Chức năng này sẽ thực hiện theo dõi quá trình làm việc và tính lương cho từng cán bộ công nhân viên trong từng tháng theo từng chức vụ và phòng ban. Mỗi chức vụ và phòng ban sẽ quy định mức lương, phụ cấp hệ số lương khác nhau. Tiền lương sẽ được tính ra hàng tháng phụ thuộc vào số ngày làm việc và mức kỷ luật. Chức năng này sẽ thực hiện cụ thể được những yêu cầu sau:

* + - In ra tổng số lương theo từng tháng của từng nhân viên.
    - Sắp xếp số tiền lương nhân viên nhận được tăng dần hoặc giảm dần.
    - Chỉnh sửa phụ cấp, mức lương quy định của từng chức vụ.
    - Thực hiện mức kỷ luật với nhân viên thông qua việc trừ tiền lương tháng bằng yêu cầu nhập số tiền trừ đi.
    - Chỉnh sửa số ngày đi làm của nhân viên bất kỳ trong tháng.

# THIẾT KẾ CẤU TRÚC DỮ LIỆU

## Phát biểu bài toán

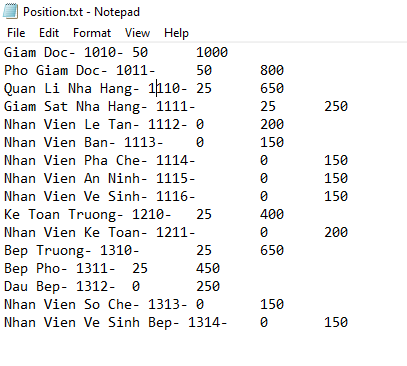
* **Input :** Đầu vào của chương trình là sẽ gồm 4 file dữ liệu
  + File lưu trữ thông tin của nhân viên ( File Person.txt)



*Hình 3.1.1: Minh họa về file lưu trữ thông tin nhân viên*

Cấu trúc lưu trữ của file sẽ được lưu theo cấu trúc sau: Tên nhân viên- số điện thoại- mã nhân viên- địa chỉ nhà- CCCD- Giới tính ( nam(1)/ nữ(0) )ngày tháng năm sinh

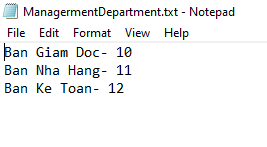
* + File lưu trữ thông tin theo vị trí ( File Position.txt )



*Hình 3.1.2: Minh họa về file lưu trữ thông tin theo chức vụ*

Cấu trúc lưu trữ của file sẽ được lưu trữ theo cấu trúc sau: Tên chức vụ- mã chức vụ- Phụ cấp theo chức vụ Tiền lương bao nhiêu ( nghìn đồng / ngày )

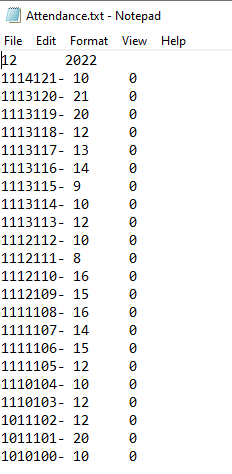
* + File lưu trữ thông tin theo phòng ban ( File ManagermetDepartment.txt )



*Hình 3.1.3: Minh họa về file lưu trữ thông tin theo phòng ban.*

Cấu trúc lưu trữ của file sẽ được lưu trữ theo cấu trúc sau: Tên phòng ban- Mã phòng ban

* + File lưu trữ theo dõi thông tin làm việc của nhân viên ( File Attendace.txt )



*Hình 3.1.4: Minh họa về file lưu trữ thông tin làm việc của nhân viên*

Cấu trúc lưu trữ của file sẽ được lưu trữ theo cấu trúc: Dòng đầu là tháng và năm bạn muốn quản lý, Thông tin tiếp theo lần lượt sẽ là mã số nhân viên- số ngày đi làm và số tiền phạt

* **Output :** Đầu ra của chương trình sẽ được lưu lại vào các file input để phục

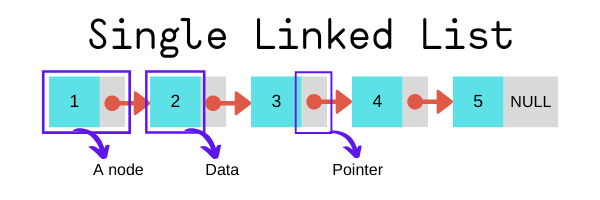
vụ cho những lần thực hiện tiếp theo

## Phân tích và ứng dụng cấu trúc dữ liệu trong hệ thống

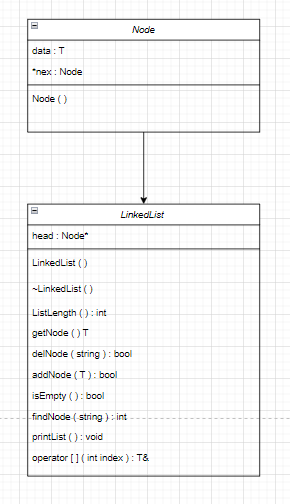
### Danh sách liên kết đơn ( Linked List )

1. **Định nghĩa danh sách liên kết đơn**

* Danh sách liên kết đơn (Single Linked List) là một cấu trúc dữ liệu động, nó là một danh sách mà mỗi phần tử đều liên kết với phần tử đúng sau nó trong danh sách. Mỗi phần tử (được gọi là một node hay nút) trong danh sách liên kết đơn là một cấu trúc có hai thành phần:
  + Thành phần dữ liệu: lưu thông tin về bản thân phần tử đó
  + Thành phần liên kết: lưu địa chỉ phần tử đúng sau trong danh sách, nếu phần tử cuối cùng thì thành phần này bằng NULL



*Hình 3.2.1: Hình minh họa danh sách liên kết đơn*



*Hình 3.2.2: Hình class LinkedList được áp dụng trong chương trình.*

* Khởi tạo Linked List:

*// Khởi tạo Node*

*Template < class T >*

*Class Node{*

*Public:*

*T data;*

*Node \* next;*

*Node( ) {*

*Next = NULL;*

*}*

*};*

*// Khởi tạo LinkedList*

*Template < class T >*

*Class LinkedList{*

*Private:*

*Node \*head;*

*Public:*

*LinkedList( ) {*

*head = NULL;*

*}*

*~LinkedList ( ) {*

*Delete [ ] head;*

*}*

*};*

1. **Lấy độ dài của LinkedList**

* Mô tả: sẽ có một biến đếm độ dài của LinkedList ( length ) cho tới khi nào con trỏ head tới NULL
* Code tính độ dài của LinkedList

*template <class T>*

*int LinkedList<T>::listLength(){*

*int length = 0;*

*Node \*temp = this->head;*

*for(temp; temp!= NULL; temp = temp->next){*

*++length;*

*}*

*return length;*

*}*

1. **Lấy dữ liệu của một Node trong LinkedList**

* Mô tả: khi truyền vào sẽ là vị trí của dữ liệu cần lấy ra trong LinkedList con trỏ sẽ thực hiện việc di chuyển tới con trỏ tiếp theo cho tới khi nào vị trí của con trỏ bằng vị trí được truyền vào và con trỏ Node không chỉ tới Null
* Code lấy một dữ liệu của một Node trong LinkedList

*template <class T>*

*T LinkedList<T>::getNode(int position){*

*int k =0;*

*Node \*temp = this->head;*

*while(k != position && temp != NULL){*

*temp = temp->next;*

*++k;*

*}*

*return temp->data;*

*}*

1. **Kiểm ra xem LinkedList có trống hay không**

* Mô tả: Sẽ kiểm tra xem Node đầu tiên có chỉ đến NULL hay không nếu có thì trả về True và ngược lại
* Code kiểm tra xem LinkedList có trông hay không ( isEmpty ):

*template <class T>*

*bool LinkedList<T>::isEmpty(){*

*if(this->head == NULL){*

*return true;*

*}return false;*

*}*

1. **Tìm kiếm trong LinkedList**

* Mô tả: Dữ liệu truyền vào sẽ là ID của chức vụ ( hay ID phòng ban ) sau đó chạy từ đầu đến cuối LinkedList khi tìm thấy ID tương ứng sẽ trả lại vị trí của ID đó trong danh sách, nếu không có sẽ trả về giá trị -1.
* Code Tìm kiếm trong LinkedList ( findNode ):

*int LinkedList<T>::findNode(string ID){*

*Node \*temp = this->head;*

*int position = 0;*

*for(temp; temp!= NULL; temp = temp->next){*

*if(temp->data.getID() == ID){*

*return position;*

*}*

*++position;*

*}*

*return -1; // node isn't in the list so return -1*

*}*

1. **Thêm Node trong LinkedList**

* Mô tả: dữ liệu đưa vào hàm sẽ là dữ liệu một Node mới cần thêm vào, dữ liệu sẽ được đưa vào đầu danh sách nếu thêm thành công sẽ trả về giá trị true ngược lại sẽ trả về giá trị false
* Code thêm Node trong LinkedList

*template <class T>*

*bool LinkedList<T>::addNode(T NewP){*

*// List is empty or new word is not in the list => add*

*if(this->isEmpty() || this->findNode(NewP.getID())){*

*Node \*temp = new Node();*

*temp->data = NewP;*

*temp->next = this->head;*

*head = temp;*

*return true;*

*}*

*// if Position has already in list => false*

*else{*

*return false;*

*}*

*}*

1. **Xoá Node trong LinkedList**

* Mô tả: dữ liệu đưa vào là ID của chức vụ ( hay ID của phòng ban ) đầu tiên sẽ tìm kiếm trong dữ liệu có ID tương ứng hay không nếu có sẽ lấy ra vị trí đó và xoá khỏi danh sách, nếu xoá thành công thì sẽ trả về giá trị true và ngược lại.
* Code xoá Node trong LinkedList:

*template <class T>*

*bool LinkedList<T>::delNode(string ID){*

*int position = this->findNode(ID);*

*int k = 0;*

*Node \*temp = this->head;*

*// if list is empty or Position is not in the list*

*if(head == NULL || position == -1)*

*return false;*

*// if the Node want to find is the first Node*

*if(this->head->next == NULL || position == 0){*

*this->head = this->head->next;*

*}*

*//other case*

*else{*

*while(temp != NULL && k != position - 1){*

*temp = temp->next;*

*++k; // go the Node which is before the Node need to delete*

*}*

*temp->next = temp->next->next;*

*}*

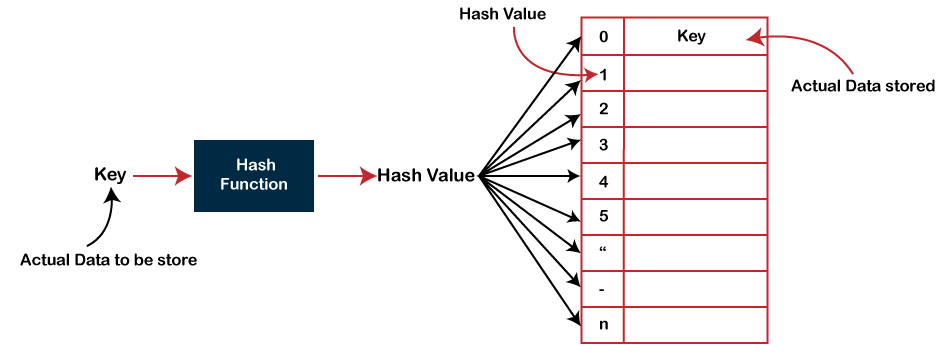
*return true;*

*}*

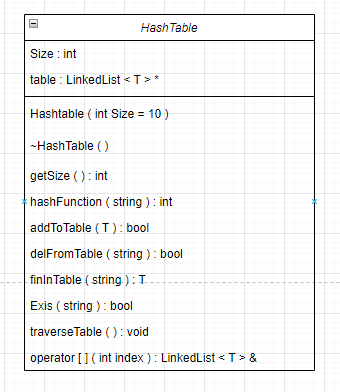
### Bảng băm ( HashTable )

1. **Định nghĩa bảng băm ( HashTable )**

* Bảng băm hay HashTable là một cấu trúc mà khi người dùng thực hiện truy xuất một phần tử qua khóa thì nó sẽ được ánh xạ vào thông qua hàm băm (Hash function).
* Quá trình ánh xạ khóa vào bảng băm được thực hiện thông qua hàm băm (HashFunction). Một bảng băm tốt cần phải có hàm băm tốt. Bảng băm là một mảng có M vị trí được đánh số từ 0 đến M – 1.

****

*Hình 3.2.3: Mô hình bảng băm (HashTable)*



*Hình 3.2.4: Class HashTable được áp dụng trong bài toán*

* Khởi tạo bảng băm ( class HashTable ):

*template <class T>*

*class HashTable{*

*private:*

*int Size;*

*LinkedList<T> \*table;*

*public:*

*HashTable(int Size = 10){*

*this->Size = size;*

*table = new LinkedList<T>[this->Size];*

*}*

*~HashTable(){*

*for(int i=0; i <this->Size;i++){*

*table[i].~LinkedList();*

*}*

*delete[] table;*

*}*

*};*

1. **Hàm băm ( Hash Fucntion )**

* Mô tả: dữ liệu đưa vào hàm là ID của chức vụ ( phòng ban ) khi xử lý xong dữ liệu đưa ra là vị trí cần Hash cần đưa vào bảng
* Code vị trí cần Hash trong bảng băm

*template <class T>*

*int HashTable<T>::hashFunction(string ID){*

*int i = (ID[0]-48)\*10 + (ID[1]-48);*

*return i%Size;*

*}*

1. **Hàm tìm kiếm trong HashTable ( findInTable Function )**

* Mô tả: tham số được truyền vào là ID của chức vụ ( phòng ban ) cần tìm, dữ liệu đưa là dữ liệu của chức vụ ( phòng ban ) đó.
* Code tìm kiếm trong HashTable theo ID trong HashTable:

*template<class T>*

*T HashTable<T>::findInTable(string ID){*

*static T temp;*

*int VT = hashFunction(ID);*

*int position = table[VT].findNode(ID);*

*// check if the list is empty or this Position isn't in the list*

*if(this->table[VT].isEmpty() || position == -1){*

*return temp;*

*}*

*else{*

*return table[VT][position];*

*}*

*}*

1. **Hàm đưa thêm dữ liệu vào HashTable ( addToTable Function )**

* Mô tả : tham số truyền vào hàm là dữ liệu của một chức vụ ( phòng ban ) mới sau khi kết thúc hàm sẽ trả về giá trị true nếu thêm thành công ( chưa tồn tại ID tương tự trong bảng ) và trả về giá trị false nếu thêm không thành công
* Code thêm dữ liệu vào HashTable:

*template <class T>*

*bool HashTable<T>::addToTable(T Input){*

*bool check;*

*int position = hashFunction(Input.getID());*

*check = table[position].addNode(Input);*

*return check;*

*}*

1. **Hàm xoá một dữ liệu từ bảng băm ( delFromTable Function )**

* Mô tả: dữ liệu đưa vào sẽ là ID của một chức vụ ( phòng ban ) dữ liệu trả về sẽ là giá trị true nếu xoá thành công dữ liệu đó khỏi bảng băm
* Code xoá một dữ liệu từ bảng băm:

*template <class T>*

*bool HashTable<T>::delFromTable(string ID){*

*bool check;*

*int index = hashFunction(ID);*

*check = table[index].delNode(ID);*

*return check;*

*}*

1. **Hàm kiểm tra một ID đã tồn tại trong bảng hay chưa ( Exis Function )**

* Mô tả: Dữ liệu đưa vào là ID của một chức vụ ( phòng ban ), dữ liệu đầu ra giá trị true nếu ID đó đã tồn tại trong bảng băm ( HashTable ) ngược lại thì trả về giá trị false
* Code kiểm tra ID đã tồn tại chưa

*template<class T>*

*bool HashTable<T>::Exis(string ID){*

*int VT = hashFunction(ID);*

*int position = table[VT].findNode(ID);*

*if(this->table[VT].isEmpty() || position == -1){*

*return false;*

*}else{*

*return true;*

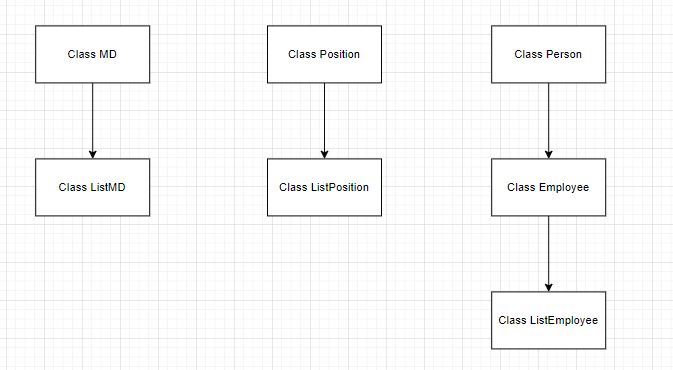
*}*

*}*

# PHÂN TÍCH HƯỚNG ĐỐI TƯỢNG VÀ TRIỂN KHAI HỆ THỐNG

## Cấu trúc hệ thống hướng đối tượng

* Chương trình gồm có gồm có 7 class:
  + **Class Person** dùng để quản lý thông tin cơ bản của nhân viên gồm có họ tên, số điện thoại địa chỉ, mã số nhân viên, CCCD, ngày sinh. Trong đó mỗi nhân viên sẽ có một mã số nhân viên khác nhau và gồm có 7 chữ số ( 2 chữ số đầu là mã số phòng ban – 4 chữ số đầu ( 2 chữ số của mã phòng ban ) là mã chức vụ – 3 số sau là thứ tự của các nhân viên )
  + **Class Employee** ( kế thừa từ class Person )dùng để quản lý thông tin của nhân viên gồm mã số nhân viên, số ngày đi làm, phụ cấp bao nhiêu % và tiền phạt của nhân viên
  + **Class ListEmployee** dùng để quản lý toàn bộ các chức năng liên quan đến danh sách nhân viên
  + **Class Position** dùng để quản lý thông tin chức vụ gồm mã chức vụ, tên chức vụ, tiền lương ( bao nhiêu nghìn đồng / ngày ), Phụ cấp của chức vụ đó ( bao nhiêu % )
  + **Class ListPosition** dùng để quản lý toàn bộ các chức năng liên quan đến danh sách chức vụ
  + **Class MD** dùng để quản lý thông tin liên quan đến phòng ban gồm mã phòng ban, tên phòng ban
  + **Class ListMD** dùng để quản lý toàn bộ chức năng liên quan đến danh sách phòng ban
* Trong đó chủ yếu quản lý trên 3 class danh sách ( ListEmployee, ListPosition và ListMD )



*Hình 4.1.1: Mô hình class được áp dụng trong chương trình*

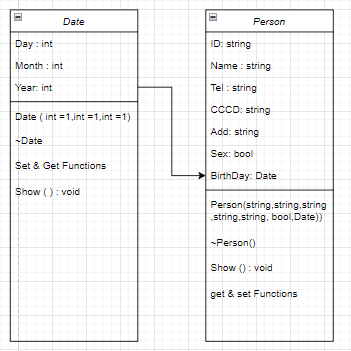
* + **Danh sách các chức năng được cài ở các List Class:** 
    - In danh sách ( có ở cả 3 danh sách – ListEmployee, ListPosition, ListMD )
    - Tìm kiếm ( có ở cả 3 danh sách – ListEmployee, ListPosition , ListMD )
    - Chỉnh sửa ( có ở cả 3 danh sách – ListEmployee, ListPosition , ListMD )
    - Thêm ( có ở cả 3 danh sách – ListEmployee, ListPosition , ListMD )
    - Xoá ( có ở cả 3 danh sách – ListEmployee, ListPosition , ListMD )
    - Sắp xếp ( chỉ có ở ListEmployee )

### Quản lý nhân viên ( class ListEmployee )

* Quản lý nhân viên gồm 3 lớp: class Person, class Employee, class ListEmployee
* Chức năng:
  + In ra danh sách lương nhân viên nhận được
  + In ra danh sách thông tin liên quan đến nhân viên
  + In ra chức vụ cũng như phòng ban nhân viên đang làm
  + Sắp xếp nhân viên theo mã số nhân viên hoặc theo tiền lương nhận được trong tháng
  + Tìm kiếm nhân viên có mã số nhân viên tương ứng
  + Thêm một nhân viên vào danh sách
  + Xoá một nhân viên ra khỏi danh sách
  + Chỉnh sửa thông tin của nhân viên trong danh sách

1. *Class Person*

* Mô tả class Person dùng để quản lý thông tin cơ bản của nhân viên gồm có họ tên, số điện thoại địa chỉ, mã số nhân viên, CCCD, ngày sinh. Trong đó mỗi nhân viên sẽ có một mã số nhân viên khác nhau và gồm có 7 chữ số ( 2 chữ số đầu là mã số phòng ban – 4 chữ số đầu ( 2 chữ số của mã phòng ban ) là mã chức vụ – 3 số sau là thứ tự của các nhân viên )

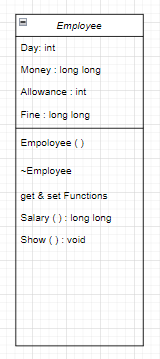


*Hình 4.1.2: Mô hình class Person được áp dụng trong chương trình.*

* Thuộc tính
  + ID ( string ) mã số nhân viên
  + Name ( string ) tên nhân viên
  + Tel ( string ) số điện thoại nhân viên
  + Add ( string ) địa chỉ nhân viên
  + CCCD ( string ) số căn cước công dân của nhân viên
  + Sex ( bool ) giới tính của nhân viên ( 1 nếu là nam và 0 nếu là nữ )
  + DateOfBirth ( Class Date ) ngày sinh của nhân viên
* Hàm chức năng trong class
  + Hàm khởi tạo và hàm huỷ
  + Hàm show ( ) dùng để in ra thông tin của nhân viên
  + Các hàm set và get tương ứng với từng thuộc tính

1. Class Employee

* Mô tả: Class Employee ( kế thừa từ class Person )dùng để quản lý thông tin của nhân viên gồm mã số nhân viên ( 7 số ), số ngày đi làm , phụ cấp ( % ) và tiền phạt của nhân viên ( nghìn đồng )

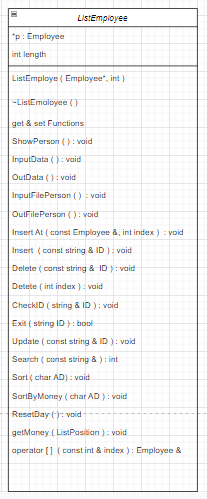


*Hình 4.1.3: Mô hình class Employee được áp dụng trong chương trình*

* Thuộc tính
  + Day ( int ) số ngày nhân viên đi làm
  + Money ( long long ) lương của nhân viên ( nghìn đông / ngày )
  + Allowance ( int ) phụ cấp nhân viên nhận được ( % )
  + Fine ( long long ) số tiền phạt của nhân viên đó ( nghìn đông )
* Hàm chức năng trong class:
  + Hàm khỏi tạo và hàm Huỷ
  + Các hàm Set và get tương ứng với từng thuộc tính
  + Hàm Salary dùng để tính số tiền của nhân viên nhận được trong tháng đó theo công thức
  + Hàm in ra thông tin của nhân viên đó ( Show Function )

1. *Class ListEmployee*

* Mô tả class ListEmployee dùng để quản lý toàn bộ các chức năng liên quan đến danh sách nhân viên



*Hình 4.1.4: Mô hình class ListEmployee được áp dụng trong chương trình*

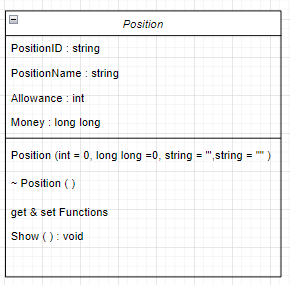
* Thuộc tính:
  + P ( Employee\* ) dùng để lưu danh sách nhân viên
  + Length ( int ) dùng để chứa độ dài của danh sách nhân viên
* Hàm chức năng của class:
  + Hàm khởi tạo và hàm huỷ
  + Các hàm set và get cho tường thuộc tính tương ứng
  + Hàm ShowPerson ( void ) dùng để in ra danh sách thông tin liên quan đến nhân viên
  + Hàm InputData & OutputData ( void ) InputData dùng để lấy dữ liệu từ file vào khi bắt đầu chương trình và OutputData dùng để đẩy dữ liệu từ chương trình ra file khi kết thúc chương trình
  + Hàm Insert và InsertAt ( void ) Insert dùng để thêm nhân viên khi nhập ID trước vào sau đó nhập thêm các thông tin còn lại sau và thêm vào cuối danh sách, InsertAt dùng để thêm một class Employee vào vị trí index đã xác định
  + Hàm Delete và DeleteAt ( void ) Delete dùng để xoá một Employee khi nhập mã số nhân viên tương ứng vào và DeleteAt dùng để xoá một nhân viên ở vị trí index đã xác định
  + Hàm CheckID ( bool ) dùng để kiểm tra mã nhân viên nhập vào đã đúng hay chưa ( 7 gồm chữ số và không được lập lại mã số nhân viên đã có trong danh sách ) sau đó trả về giá trị true/false
  + Hàm Exit ( bool ) dùng để kiểm tra xem mã nhân viên đó đã tồn tại trong danh sách chưa nếu đã tồn tại trả về giá trị true và ngược lại trả về giá trị false
  + Hàm Update ( void ) dùng để Update dữ liệu của nhân viên khi nhập vào một mã số nhân viên đã có trong danh sách
  + Hàm Search ( int ) dùng để tìm một mã số nhân viên đã nhập vào và trả về giá vị trí tương tứng trong danh sách
  + Hàm Sort và SortByMoney ( void ) Sort dùng để sắp xếp lại danh sách theo thứ tự tăng dần hoăc giảm dần của mã số nhân viên và SortByMoney dùng để sắp xếp lại danh sách theo thứ tự tăng dần hoặc giảm dần theo lương của nhân viên nhận được
  + Hàm ResetDay ( void ) dùng để đưa tất cả ngày đi làm của nhân viên trở về 0
  + Hàm getMoney dùng để so sánh với danh sách chức vụ có mã số tương ứng và lấy số lương ( nghìn đông / ngày ) và phụ cấp ( % )

### Quản lý chức vụ ( class ListPosition )

* Quản lý chức vụ gồm 2 lớp : class Position và class ListPosition
* Chức năng
  + In ra các chức vụ có trong danh sách
  + Thêm vào danh sách một chức vụ mới
  + Xoá chức vụ ra khỏi danh sách theo ID chức vụ
  + Tìm kiếm trong danh sách theo ID chức vụ
  + Chỉnh sửa thông tin của chức vụ

1. Class Position

* Mô tả Class Position dùng để quản lý thông tin chức vụ gồm mã chức vụ, tên chức vụ, tiền lương ( bao nhiêu nghìn đồng / ngày ), Phụ cấp của chức vụ đó ( bao nhiêu % )

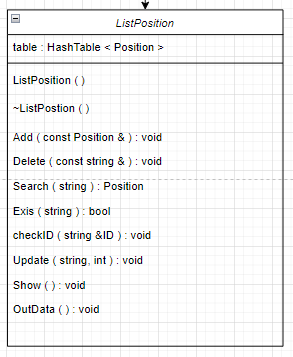


*Hình 4.1.5: Mô hình class Position được áp dụng trong chương trình*

* Thuộc tính
  + PositionID ( string ) mã chức vụ ( gồm 4 chữ số, 2 chữ số đầu là phòng ban của chức vụ đó )
  + PositionName ( string ) tên chức vụ
  + Allowance ( int ) là phụ cấp mà nhân viên nào làm chức vụ đó nhận được
  + Money ( long long ) là số tiền mà chức vụ đó nhận được trong 1 ngày ( đơn vị nghìn đồng / ngày )
* Hàm chức năng của class
  + Hàm khỏi tạo và hàm huỷ
  + Các hàm set và get tương ứng với từng thuộc tính
  + Hàm show ( void ) dùng để in ra thông tin của chức vụ đó

1. Class ListPositition

* Mô tả dùng để quản lý toàn bộ các chức năng liên quan đến danh sách chức vụ



*Hình 4.1.6: Mô hình class ListPosition được áp dụng trong chương trình*

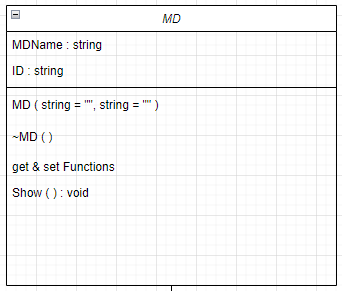
* Thuộc tính
  + Table ( HashTable < Position > ) là một hash table mang kiểu dữ liệu template là class Position dùng để chứa danh sách Position
* Hàm chức năng của class
  + Hàm khởi tạo và hàm huỷ tương ứng với từng thuộc tính
  + Hàm Add ( void ) dữ liệu truyền vào là một Position sẽ được thêm vào thuộc tính Table của class
  + Hàm Delete ( void ) dữ liệu truyền vào là ID của một chức vụ có trong danh sách và sẽ được xoá khỏi thuộc tính table của class
  + Hàm Search ( class Position ) dữ liệu truyền vào là ID của chức vụ có trong danh sách và sẽ tiến hành tìm trả về giá trị Position tương ứng
  + Hàm Exis ( bool ) dữ liệu truyền vào là ID của một chức vụ và kiểm tra xem chức vụ đó đã tồn tại trong danh sách chưa nếu đã tồn tại thì trả về giá trị true và ngược lại trả về giá trị false
  + Hàm checkID ( bool ) dữ liệu truyền vào là ID của một chức vụ và sẽ kiểm tra xem ID của chức vụ đó có đúng chưa ( đủ 4 chữ số và chưa tồn tại trong danh sách ) nếu đúng thì trả về giá trị true và ngược lại trả về giá trị false
  + Hàm Update ( void ) truyền vào là mã số chức năng cần chỉnh sửa thông tin và sự lựa chọn ( 1 là sửa mã số chức vụ, 2 là sửa tên chức vụ, 3 là sửa tiền lương , 4 là sửa phụ cấp )
  + Hàm Show ( void ) in ra danh sách tất cả các chức vụ có trong danh sách
  + Hàm OutData ( void ) dùng để đẩy dữ liệu từ chương trình ra file sau khi kết thúc chương trình để dùng cho lần chạy tiếp theo

### Quản lý phòng ban ( class ListMD )

* Quản lý phòng ban gồm 2 lớp : class MD và class ListMD
* Chức năng
  + In ra các phòng ban có trong danh sách
  + Thêm vào danh sách một phòng ban mới
  + Xoá phòng ban ra khỏi danh sách theo ID phòng ban
  + Tìm kiếm trong danh sách theo ID phòng ban
  + Chỉnh sửa thông tin của phòng ban

1. Class MD

* Mô tả dùng để quản lý thông tin liên quan đến phòng ban gồm mã phòng ban, tên phòng ban

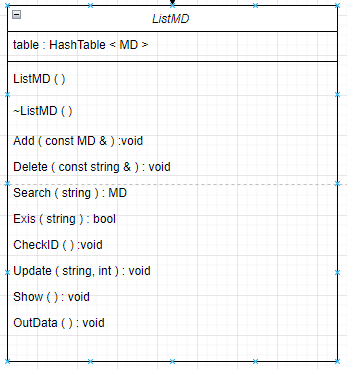


*Hình 4.1.7: Mô tả class MD được áp dụng trong chương trình*

* Thuộc tính
  + MDName ( string ) tên phòng ban
  + ID ( string ) là mã phòng ban ( gồm 2 số )
* Hàm chức năng của class
  + Hàm khởi tạo và hàm huỷ
  + Các hàm get và set tương ứng
  + Hàm Show ( void ) dùng để in thông tin của phòng ban ra

1. Class ListMD

* Mô tả dùng để quản lý toàn bộ chức năng liên quan đến danh sách phòng ban

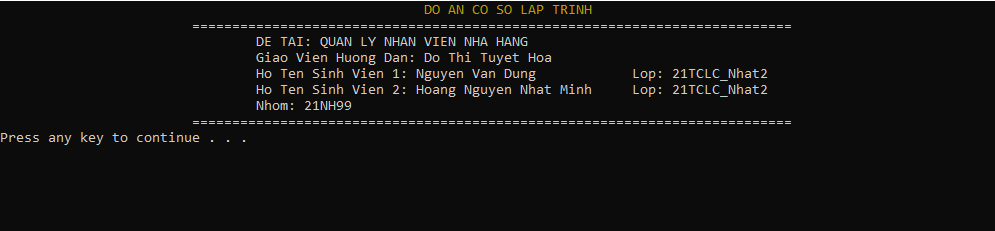


*Hình 4.1.8: Mô tả class ListMD được áp dụng trong chương trình*

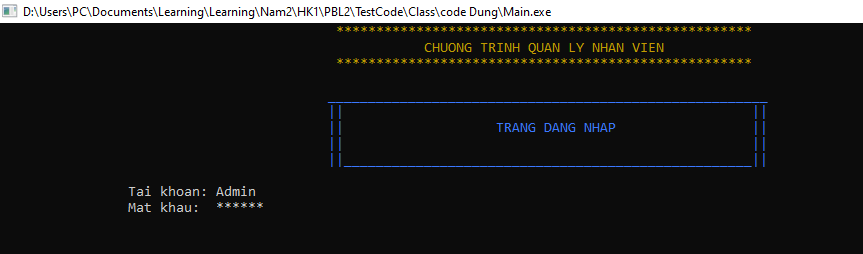
* Thuộc tính
  + Table ( HashTable < MD> ) là một hash table mang kiểu dữ liệu template là class MD dùng để chứa danh sách MD
* Hàm chức năng của class
  + Hàm Add ( void ) dữ liệu truyền vào là một class MD sẽ được thêm vào thuộc tính Table của class
  + Hàm Delete ( void ) dữ liệu truyền vào là ID của phòng ban có trong danh sách và sẽ được xoá khỏi thuộc tính table của class
  + Hàm Search ( class Position ) dữ liệu truyền vào là ID của phòng ban có trong danh sách và sẽ tiến hành tìm trả về giá trị class MD tương ứng
  + Hàm Exis ( bool ) dữ liệu truyền vào là ID của một phòng ban và kiểm tra xem phòng ban đó đã tồn tại trong danh sách chưa nếu đã tồn tại thì trả về giá trị true và ngược lại trả về giá trị false
  + Hàm checkID ( bool ) dữ liệu truyền vào là ID của một phòng ban và sẽ kiểm tra xem ID của phòng ban đó có đúng chưa ( đủ 2 chữ số và chưa tồn tại trong danh sách ) nếu đúng thì trả về giá trị true và ngược lại trả về giá trị false
  + Hàm Update ( void ) truyền vào là mã số phòng ban cần chỉnh sửa thông tin và sự lựa chọn ( 1 là sửa mã số phòng ban, 2 là sửa tên phòng ban)
  + Hàm Show ( void ) in ra danh sách tất cả các phòng ban có trong danh sách
  + Hàm OutData ( void ) dùng để đẩy dữ liệu từ chương trình ra file sau khi kết thúc chương trình để dùng cho lần chạy tiếp theo

## Kết quả

### Giao diện chính của chương trình



*Hình 4.2.1: Giao diện chương trình khởi đầu*



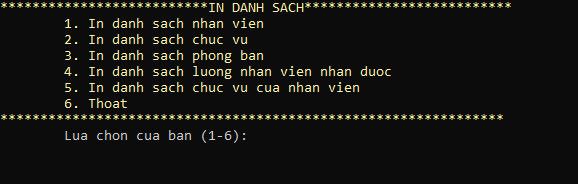
*Hình 4.2.2: Giao diện đăng nhập của chương trình*



*Hình 4.2.3: Giao diện của sổ lựa chọn các chức năng của chương trình*

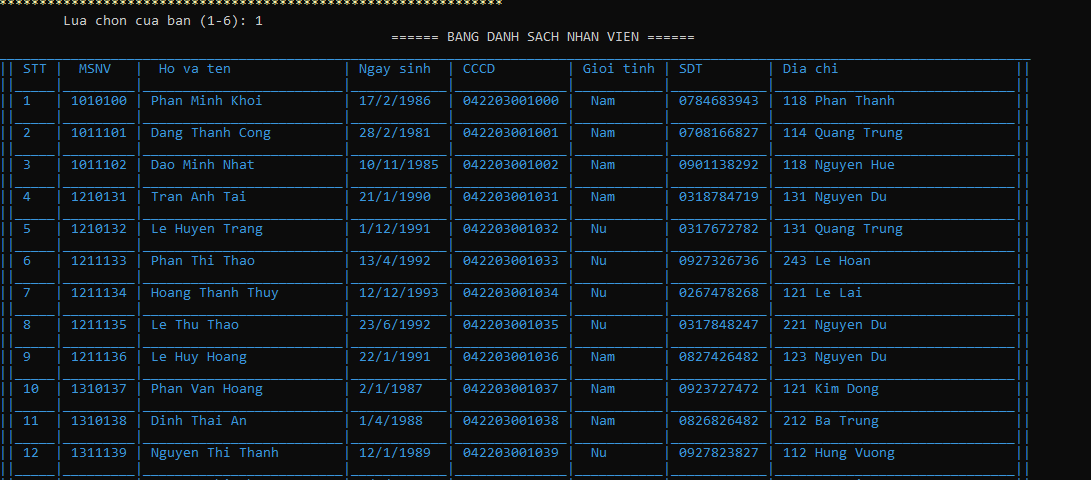
### Kết quả thực thi của chương trình

1. Chức năng in danh sách



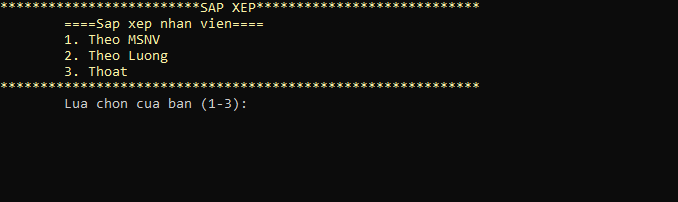
*Hình 4.2.4: Giao diện chức năng in danh sách*

* Mô tả : trong này gồm các chức năng là in danh sách nhân viên ( số 1 ), in danh sách các chức vụ hiện có trong nhà hàng ( số 2 ), in danh sách phòng ban hiện có trong nhà hàng ( số 3 ), Tổng kết và in ra danh sách lương của từng nhân viên nhận được ( số 4 ), In ra danh sách chức vụ và phòng ban của nhân viên tương ứng ( số 5 )
* Ví dụ : 1. In danh sách nhân viên hiện có trong nhà hàng



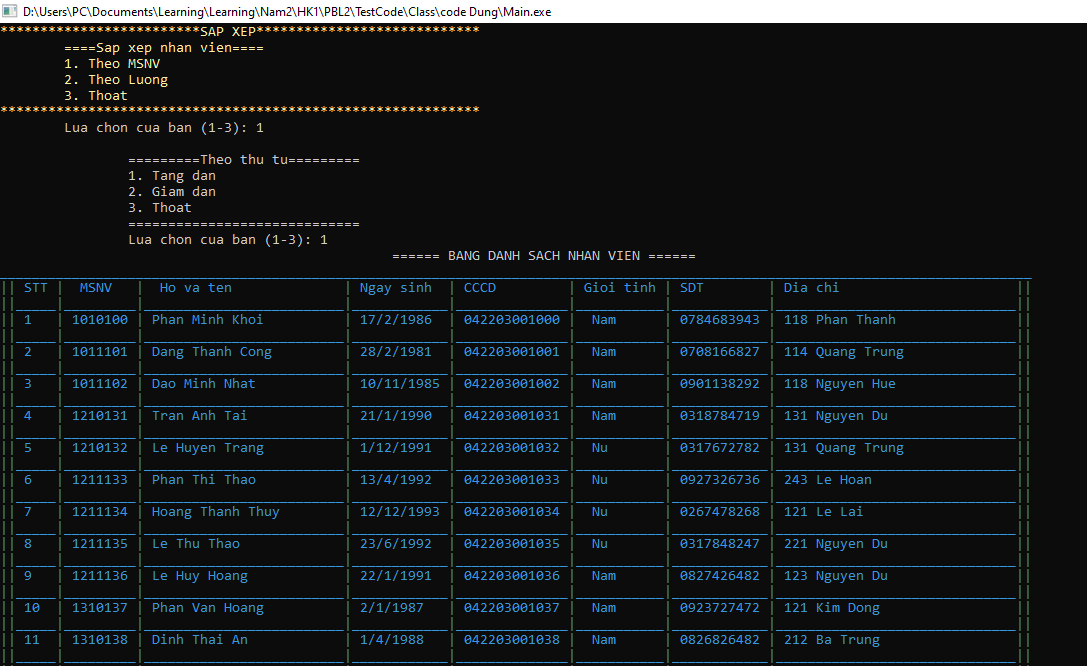
*Hình 4.2.5: In danh sách nhân viên có trong nhà hàng*

1. Chức năng sắp xếp



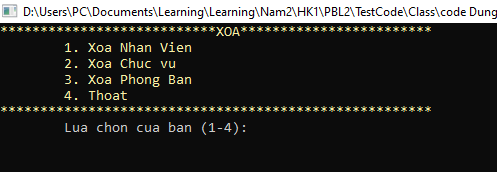
*Hình 4.2.6: Giao diện chức năng sắp xếp*

* Mô tả: trong này gồm có các chức năng là sắp xếp theo mã số nhân viên ( số 1) và sắp xếp theo lương mà nhân viên nhận được ( số 2 ) theo thứ tự tăng dần hoặc giảm dần
* Ví dụ: Sắp xếp danh sách theo mã số nhân viên ( số 1 ) theo thứ tự tăng dần



*Hình 4.2.7: Sắp xếp nhân viên theo thứ tự tăng dần của mã số nhân viên*

1. Chức năng xoá trong chương trình



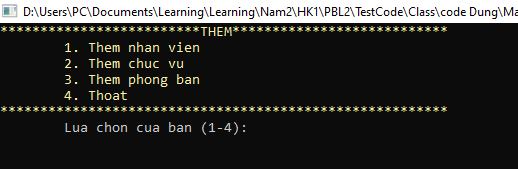
*Hình 4.2.8: Giao diện chức năng xóa*

* Mô tả: trong chương trình gồm các chức năng xoá nhân viên ra khỏi danh sách ( số 1 ), xoá chức vụ ra khỏi danh sách ( số 2 ), xoá phòng ban ra khỏi danh sách ( số 3 ).
* Ví dụ: Xoá 1 phòng ban ra khỏi danh sách ( số 3)



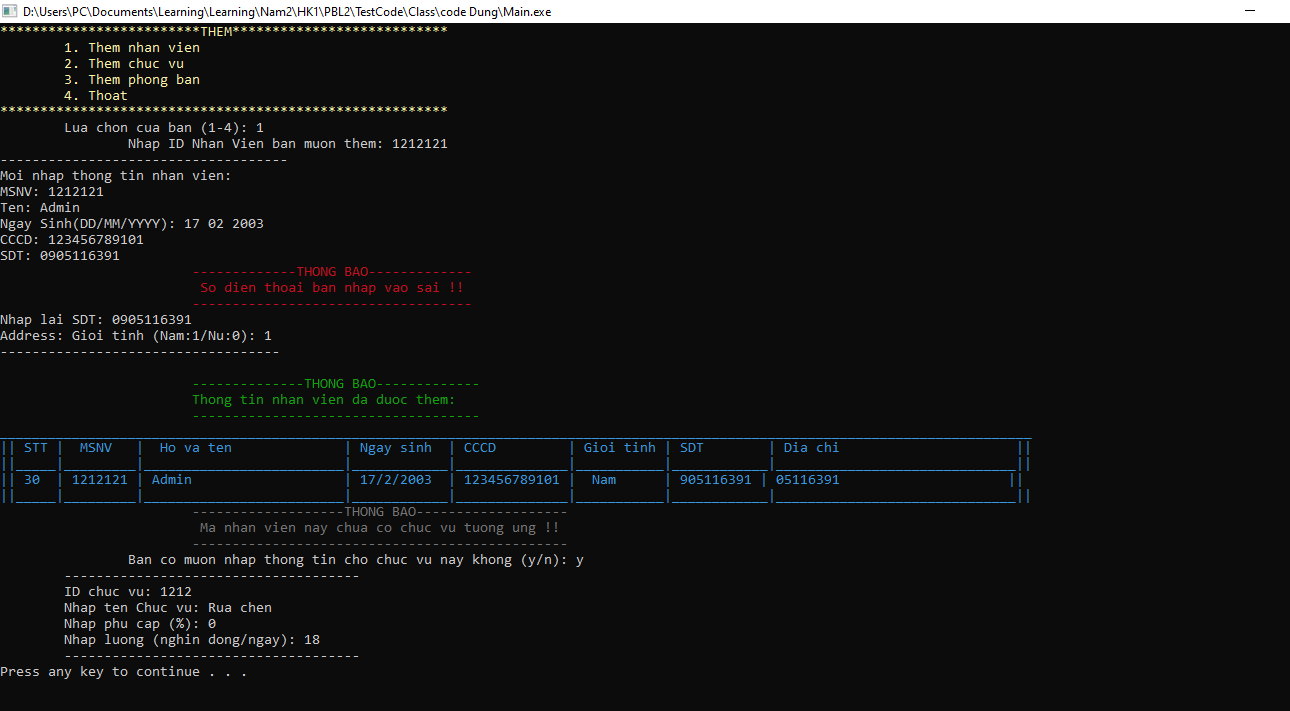
*Hình 4.2.9: Xóa một phòng ban ra khỏi danh sách*

1. Chức năng thêm trong chương trình



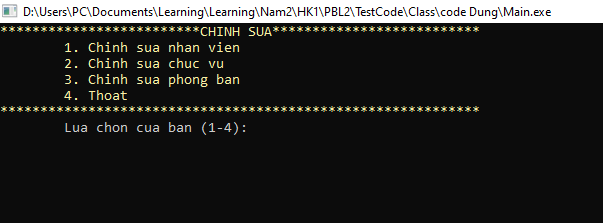
*Hình 4.2.10: Giao diện chức năng thêm trong danh sách*

* Mô tả: trong chương trình gồm các chức năng thêm một nhân viên vào trong danh sách ( số 1 ), thêm chức vụ vào trong danh sách ( số 2 ), thêm phòng ban vào trong danh sách ( số 3 ).
* Ví dụ: Thêm một nhân viên ( số 1 ) vào trong danh sách



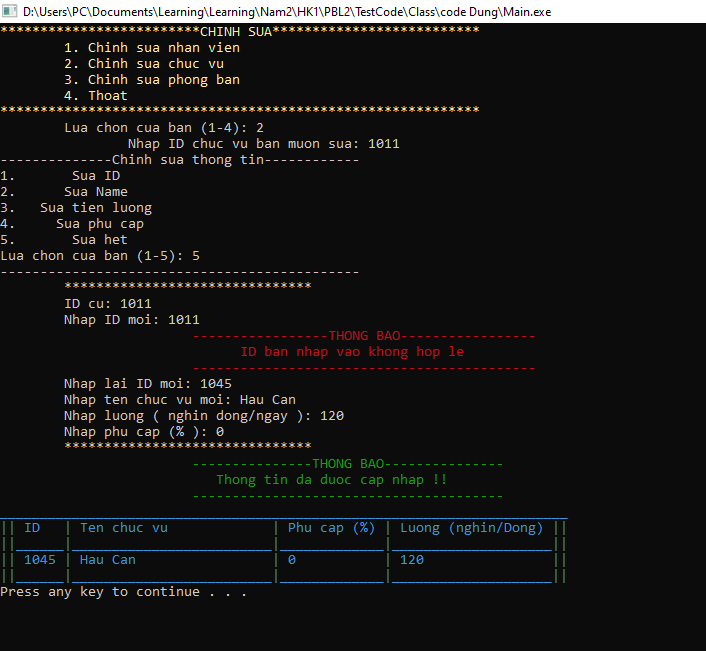
*Hình 4.2.11: Thêm một nhân viên mới vào trong danh sách*

1. Chức năng



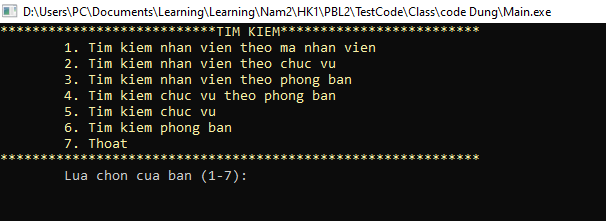
*Hình 4.2.12: Giao diện chỉnh sửa thông tin trong danh sách*

* *­*Mô tả : gồm các chức năng chỉnh sửa thông tin nhân viên ( số 1 ), chức năng chỉnh sửa thông tin chức vụ (số 2 ), chức năng chỉnh sửa thông tin phòng ban ( số 3 ) .
* Ví dụ: Chỉnh sửa thông tin chức vụ ( số 2 )



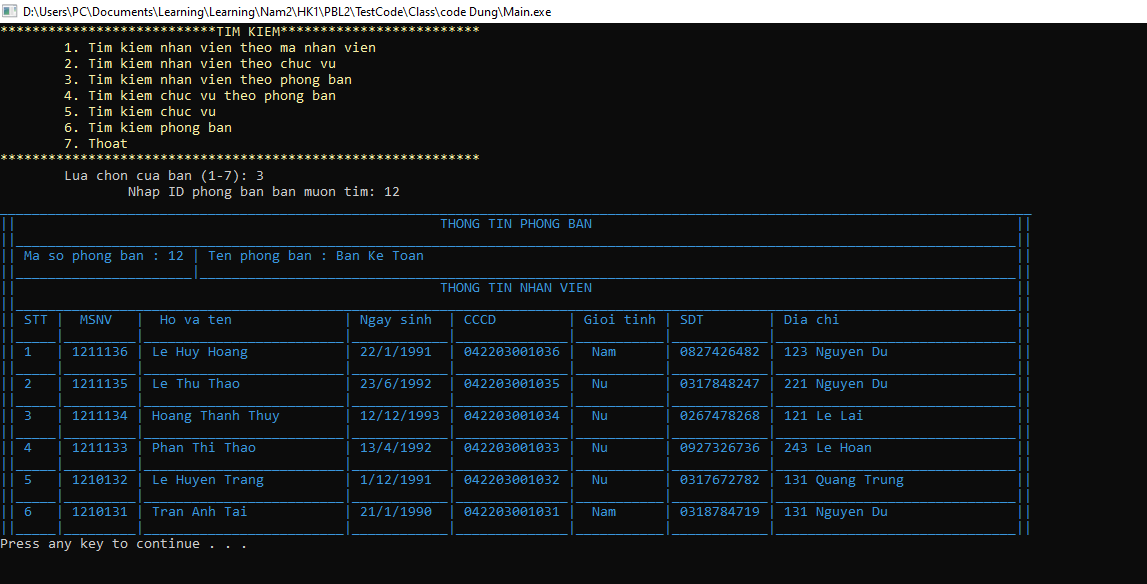
*Hình 4.2.13: Chỉnh sửa thông tin của một nhân viên trong danh sách*

1. Chức năng tìm kiếm



*Hình 4.2.14: Giao diện chức năng tìm kiếm trong chương trình*

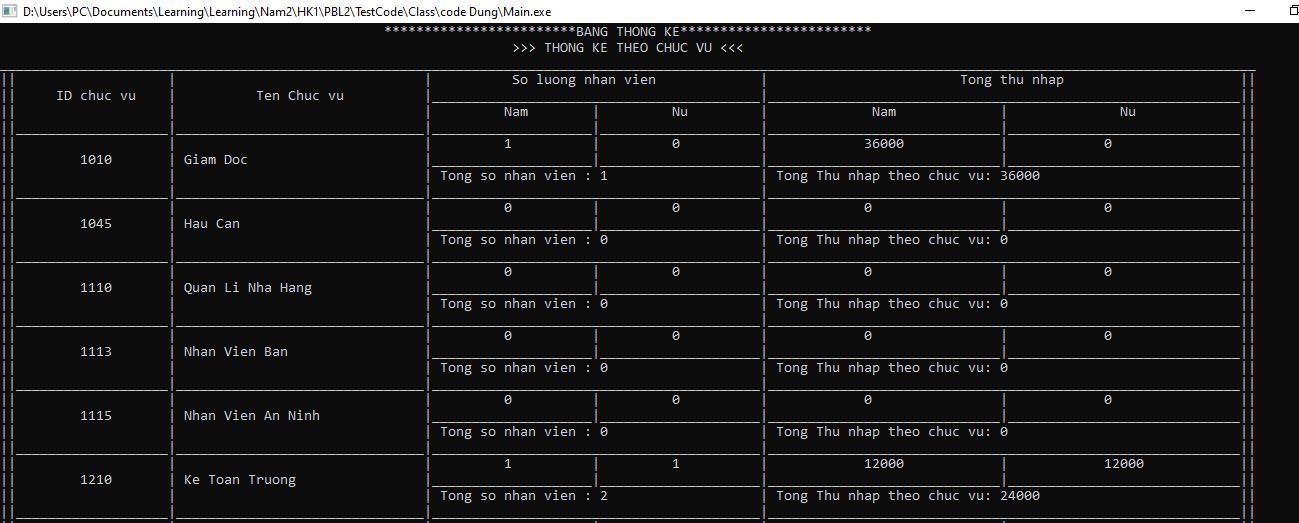
* Mô tả: trong này gồm có các chức năng tìm kiếm nhân viên theo mã nhân viên ( số 1 ) , tìm kiếm nhân viên theo mã chức vụ ( số 2 ), Tìm kiếm nhân viên theo mã phòng ban ( số 3 ), tìm kiếm chức vụ theo mã phòng ban ( số 4 ), tìm kiếm chức vụ ( số 5) , tìm kiếm phòng ban ( số 6 )
* Ví dụ tìm kiếm nhân viên theo phòng ban ( số 3 )



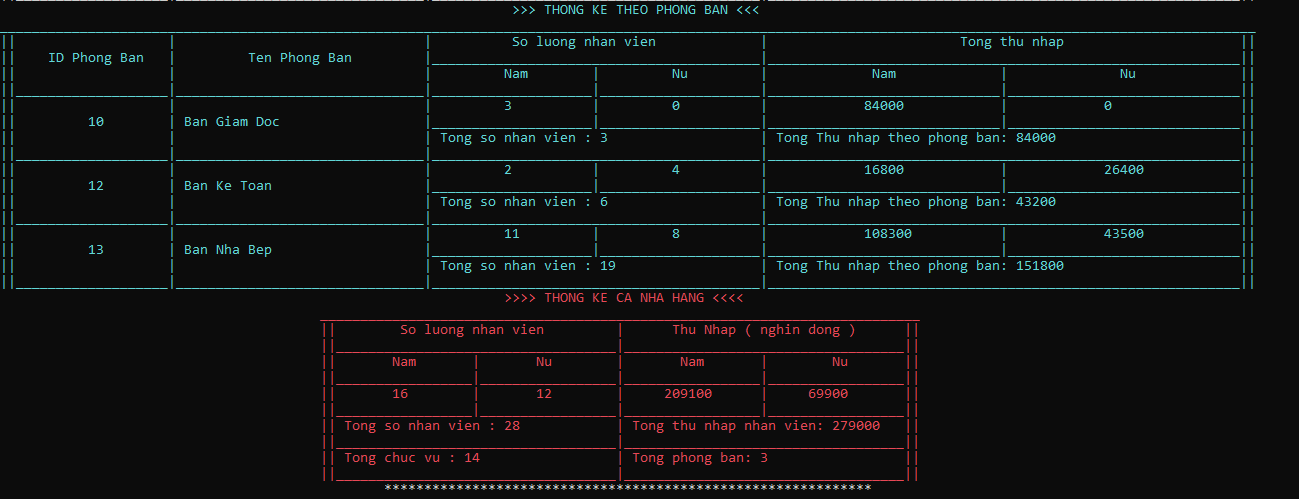
*Hình 4.2.15: Tìm kiếm nhân viên mã phòng ban.*

1. Chức năng thống kê

* Mô tả : thống kê số nhân viên nam số nhân viên nữ, tổng số thu nhập theo nam và nữ, số nhân viên và tổng số



*Hình 4.2.16: Giao diện thông kê nhân viên*



*Hình 4.2.17: Chức năng thống kê được áp dụng trong chương trình*

### Nhận xét

Chương trình hiển thị kết quả dựa trên dữ liệu thông tin nhân viên có sẵn trong file txt được tích hợp trong chương trình. Các chức năng truy cập, chèn, xóa, chỉnh sửa, thêm của chương trình sẽ được cập nhật trực tiếp vào file txt nhằm bảo đảm tính toàn vẹn, chính xác của dữ liệu thông tin nhân viên.

Chương trình chỉ mới chạy thực hiện trên nguồn dữ liệu thông tin nhân viên còn hạn chế, chưa sát với thực tế bên ngoài nên vẫn còn một số chức năng còn hạn chế về khả năng truy cập, thêm, xóa, chỉnh sửa một lượng lớn dữ liệu thông tin nhân viên. Để giải quyết vấn đề trên chương trình cần tích hợp sử dụng cơ sở dữ liệu để quản lý, các cấu trúc dữ liệu nhằm nâng cao tốc độ truy cập, chèn, xóa, chỉnh sửa dữ liệu thông tin nhân viên bảo đảm tính toàn vẹn, chính xác, có độ bảo mật cao, tránh gây tràn dữ liệu.

# KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

## Kết luận

### Tóm tắt những kết quả đạt được.

Chương trình quản lý nhân viên đã thực hiện được những chức năng quản lý với mong muốn tạo tiền đề để xây dựng lên một phần mềm quản lý nhân viên hoàn chỉnh có giao diện người dùng ứng dụng trong thực tế.

Chương trình quản lý nhân viên đã thực hiện được những công việc sau:

* + - * + In ra danh sách thông tin nhân viên.
        + Sắp xếp thông tin nhân viên.
        + Xóa, thêm, cập nhật thông tin nhân viên, chức vụ và phòng ban.
        + Tra cứu thông tin nhân viên, chức vụ, phòng ban làm việc.
        + Thông kế thông tin nhân viên theo từng chức vụ trong từng phòng ban.
        + Quan lý số tiền lương của nhân viên theo từng chức vụ trong từng phòng ban thông qua số ngày đi làm trong tháng, số tiền bị phạt, phụ cấp, và mức lương quy định theo từng chức vụ.

### Những mặt còn hạn chế.

* + - * + Giao diện của chương trình chưa được thiết kế đẹp.
        + Các chức năng chưa có cửa sổ giao diện đáp ứng nhu cầu theo dõi, đánh giá của người dùng.
        + Chưa áp dụng bộ gõ tiếng Việt cho chương trình.
        + Thao tác các điều kiện cho người dùng sử dụng các chức năng chưa tốt.

## Hướng phát triển

* + - Phát triển giao diện người dùng.
    - Phân quyền cho người quản lý chương trình.
    - Bổ sung một số chức năng quản lý lương thưởng cho nhân viên vào những dịp nghỉ lễ, quản lý bảo hiểm y tế của nhân viên,...

TÀI LIỆU THAM KHẢO

[1] <https://www.youtube.com/watch?v=dMyhU07qcgs&t=2390>s

[2] <https://topdev.vn/blog/bang-bam-trong-c/>

[3] <https://www.hoteljob.vn/tin-tuc/so-do-va-co-cau-nhan-su-nha-hang-ban-can-biet>

[4] <https://doan.edu.vn/do-an/de-tai-phan-mem-quan-ly-nhan-su-39270/>

[5] <https://www.youtube.com/watch?v=xMp7wz1Rwm4>

[6] <https://lhchuong.wordpress.com/category/data-structures-algorithms/>

[7] <https://www.softwaretestinghelp.com/linked-list/>